Amendments to the Claims

1. (Currently Amended) A protocol performance test method ~~for testing a computer system to be operated in a multi-computer environment~~, comprising:

measuring performance of a multi-computer communication protocol on a single computer system, wherein said single computer system emulates a server and a client, and wherein measuring said performance of said multi-computer communication protocol on said single computer system includes,

executing server code ~~at a~~ on said single computer system ~~under test~~;

executing client code ~~at~~ on said single computer system ~~under test~~; and

~~calculating~~ measuring performance data for said single computer system ~~under test~~.

2. (Currently Amended) The method of claim 1 wherein said server code and said client code include a number of threads, the method further comprising:

tracking an execution time for each of said threads by a processor in said single computer system ~~under test~~; and

tracking a number of transactions completed between the execution of said server code and the execution of said client code wherein said performance data is based on said number of transactions completed over a period of time.

3. (Currently Amended) A protocol performance test method ~~for testing a computer system to be operated in a multi-computer environment~~, comprising:

operating a computer system under test as a server and a client;

executing server code ~~at a~~ on said computer system under test according to a ~~multicomputer~~ multi-computer communication protocol;

executing client code on said computer system under test according to said ~~multicomputer~~ multi-computer communication protocol; and

~~calculating~~ determining performance data for said computer system under test ~~operating as one of a server and a client~~.

4. (Currently Amended) The method of claim 3 wherein said server code and said client code includes a number of threads, the method further comprising:

tracking an execution time for each of said threads by a processor in said computer system under test.

5. (Currently Amended) The method of claim 4 wherein said ~~multicomputer~~ multi-computer communication protocol defines transactions between said server and said client, the method further comprising:

tracking a number of transactions completed between the execution of said server code and the execution of said client code.

6. (Original) The method of claim 5 wherein said performance data is based on said number of transactions completed over a period of time.

-5-

Yeh *et al.*
Appl. No. 09/851,725

7. (Currently Amended) The method of claim 6 wherein said performance data is based on said number of transactions completed over said period of time modified by a scaling factor.

8. (Currently Amended) The method of claim 7 wherein said scaling factor ~~is~~ comprises a total execution time for both client and server threads divided by one of an execution time for said server threads and an execution time for said client threads.

9. (Currently Amended) A set of instructions residing in a storage medium, said set of instructions capable of being executed by a processor to implement a protocol performance test method ~~for testing a computer sytem to be operated in a multi-computer environment~~, the method comprising:

measuring performance of a multi-computer communication protocol on a single computer system, wherein said single computer system emulates a server and a client, and wherein measuring said performance of said multi-computer communication protocol on said single computer system includes,

executing server code ~~at a~~ on said single computer system ~~under test~~;
executing client code ~~at~~ on said single computer system ~~under test~~; and
~~calculating~~ measuring performance data for said single computer system ~~under test~~.

10. (Currently Amended) The set of instructions of claim 9, wherein said server code and said client code include a number of threads, the method further comprising:

tracking an execution time for each of said threads by a processor in said single computer system ~~under test~~; and

tracking a number of transactions completed between the execution of said server code and the execution of said client code wherein said performance data is based on said number of transactions completed over a period of time.

11. (Currently Amended) A set of instructions residing in a storage medium, said set of instructions capable of being executed by a processor to implement a protocol performance test method ~~for testing a computer sytem to be operated in a multi-computer environment~~, the method comprising:

operating a computer system under test as a server and a client;
executing server code ~~at a~~ on said computer system under test according to a ~~multicomputer~~ multi-computer communication protocol;
executing client code on said computer system under test according to said ~~multicomputer~~ multi-computer communication protocol; and
~~calculating~~ determining performance data for said computer system under test ~~operating as one of a server and a client~~.

12. (Currently Amended) The set of instructions of claim 11 wherein said server code and said client code includes a number of threads, the method further comprising:

tracking an execution time for each of said threads by a processor in said computer system under test.

-6-

Yeh *et al.*
Appl. No. 09/851,725

13. (Currently Amended) The set of instructions of claim 12 wherein said ~~multicomputer~~ multi-computer communication protocol defines transactions between said server and said client, the method further comprising:

tracking a number of transactions completed between the execution of <u>said</u> server code and the execution of <u>said</u> client code.

14. (Original) The set of instructions of claim 13 wherein said performance data is based on said number of transactions completed over a period of time.

15. (Currently Amended) The set of instructions of claim 14 wherein said performance data is based on said number of transaction<u>s</u> completed over said period of time modified by a scaling factor.

16. (Currently Amended) The set of instructions of claim 15 wherein said scaling factor ~~is~~ <u>comprises</u> a total execution time for both client and server threads divided by one of an execution time for said server threads and an execution time for said client threads.

17. (Currently Amended) A computer system under test to be operated in a multi-computer environment, comprising:

a processor to execute server code and client code ~~at~~ <u>on</u> said computer system under test according to a ~~multicomputer~~ <u>multi-computer</u> communication protocol, said computer system ~~uner~~ <u>under</u> test to ~~calculate~~ <u>determine</u> performance data for said <u>multi-computer communication protocol, said</u> computer system under test operating as ~~one of~~ a server and a client.

18. (Currently Amended) The computer system of claim 17 wherein said server code and <u>said</u> client code includes a number of threads, ~~and the~~ <u>said</u> computer system under test ~~is~~ to track an execution time for each of said threads by said processor.

19. (Currently Amended) The computer system of claim 18 wherein said ~~multicomputer~~ <u>multi-computer</u> communication protocol defines transactions between said server and said client, ~~and the~~ <u>said</u> computer system under test ~~is~~ to track a number of transactions completed between the execution of <u>said</u> server code and the execution of <u>said</u> client code.

20. (Original) The computer system of claim 19 wherein said performance data is based on said number of transactions completed over a period of time.

21. (Currently Amended) The computer system of claim 20 wherein said performance data is based on said number of transaction<u>s</u> completed over said period of time modified by a scaling factor.

22. (Currently Amended) The computer system of claim 21 wherein said scaling factor ~~is~~ <u>comprises</u> a total execution time for both client and server threads divided by

-7-

Yeh *et al.*
Appl. No. 09/851,725

one of an execution time for said server threads and an execution time for said client threads.

23. (New) The method of claim 1, wherein said server code comprises a number of server threads, said client code comprises a number of client threads, and wherein said method further comprises executing scheduler code, said scheduler code comprising a number of scheduler threads, said scheduler threads for coordinating communication of data between said client threads and said server threads.

24. (New) The method of claim 23, wherein executing said scheduler code includes interfacing with a queue to store data packets to be transferred to a client thread or a server thread.

25. (New) The set of instructions of claim 9, wherein said server code comprises a number of server threads, said client code comprises a number of client threads, and wherein said method further comprises executing scheduler code, said scheduler code comprising a number of scheduler threads, said scheduler threads for coordinating communication of data between said client threads and said server threads.

26. (New) The set of instructions of claim 25, wherein executing said scheduler code includes interfacing with a queue to store data packets to be transferred to a client thread or a server thread.